Componente P2P para o Middleware Ginga

Armando Biagioni Neto
Institute de Ciências Matemáticas e de Computação
Universidade de São Paulo
São Carlos - SP, Brazil
armandonet@grad.icmc.usp.br

Maria da Graça Campos Pimentel (Orientadora)
Institute de Ciências Matemáticas e de Computação
Universidade de São Paulo
São Carlos - SP, Brazil
mgp@icmc.usp.br

ABSTRACT

Ginga is the middleware for SBTVD, the Brazilian Digital TV System. The purpuse of this study is to extend Ginga, creating a software component that enables it to communicate in a peer-to-peer network, and exposing an API that developers can use to exchange files and messages in the Ginga ecossystem without the need to explore the details of peer-to-peer communication. This required understanding of the workings and structure of the middleware and development tools, and also the investigation of possible protocols for file and message exchange in the new component. This work resulted in the adoption of the XMPP-Jingle protocol in a prototype component, its specification and a test case, showing the middleware's proneness to be componentized for use in non-conventional applications.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms

Design, Experimentation

Keywords

Digital Television, Peer-to-Peer, SBTVD, Ginga Middleware, XMPP, Jingle

1. INTRODUÇÃO

A televisão digital está sendo implementada em todo o território nacional, com isso, novos recursos são disponibilizados, entre eles a possibilidade de interação com os programas transmitidos, a partir de aplicações desenvolvidas para esse fim. A comunicação ponto-a-ponto entre os espectadores permite a colaboração e troca de arquivos, um novo recurso disponível a ser explorado pelos desenvolvedores das aplicações para TV Digital Interativa (TVDi).

WebMedia 2010 Belo Horizonte, MG Brasil

Ginga é o nome do Middleware Aberto do Sistema Brasileiro de TV Digital (SBTVD). A Rede Nacional de Pesquisa (RNP) é a incubadora do programa Centro de Pesquisa e Desenvolvimento em Tecnologias Digitais para Informação e Comunicação (CTIC) [1], a qual propôs uma série de tarefas para o desenvolvimento do *middleware* dentro do projeto intitulado: *GingaFrEvo e GingaRAP - Evolução do Middleware Ginga para Múltiplas Plataformas (Componentização) e Ferramentas para Desenvolvimento e Distribuição de Aplicações Declarativas.* O projeto é subdivido em GingaRAP, destinado ao suporte a autoria de aplicações, e GingaFrEvo, um *framework* de evolução da tecnologia Ginga, e é realizado em conjunto com várias universidades e coordenado pelos laboratórios Telemídia da Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) e o Laboratório de Aplicações de Vídeo Digital (LAViD) da Universidade Federal da Paraíba (UFPB) [6].

Inserido no GingaFrEvo está o GingaAiyê, especialização do Ginga-CC (*Ginga-Common Core*) para aplicações não convencionais. Ginga-CC é o nome dado a um dos grandes módulos do *middleware* Ginga, o Núcleo Comum, que será abordado mais a frente neste trabalho. Uma das motivações para o GingaAiyê é a demanda pela padronização do *middleware* Ginga para plataformas de IPTV, Internet TV e P2P TV, abrindo a possibilidade para que os usuários realizem atividades colaborativas por meio da Internet [6].

Nos computadores a troca de arquivo entre usuários é um elemento essencial para a colaboração, seja por e-mail ou redes de compartilhamento. Com atenção a esse potencial, o desenvolvimento da interatividade da TV Digital deve oferecer tal recurso, de modo que desenvolvedores terão mais uma importante ferramenta, que possibilitará o surgimento de aplicações mais interessantes para a televisão. Esse trabalho tem como objetivo descrever a especificação e implementação iniciais de um componente P2P que proporcione a colaboração entre usuários via servidores apropriados.

O documento está estruturado da seguinte forma: a Seção 2 introduz a área explorada, protocolo e biblioteca utilizada no trabalho. A Seção 3 mostra detalhes da arquitetura do *middleware* Ginga e como o trabalho foi realizado. Por fim, a Seção 4 apresenta o resultado obtido.

2. SISTEMAS DISTRIBUÍDO P2P

Sistemas distribuídos são classificados em Cliente/Servidor e *Peer-to-Peer*. Um sistema Cliente/Servidor consiste em um sistema de alta performance, o servidor, e vários outros

[1]http://www.rnp.br/pd/ctic.html

sistemas de performance mais baixa, os clientes. O servidor é o provedor de conteúdo e serviço. Um cliente requisita por conteúdo ou execução de serviços, sem compartilhar qualquer de seus recursos. Recursos podem ser poder de processamento, capacidade de armazenamento de dados, impressoras, entre outros; serviços são compartilhamento de arquivos, por exemplo.

Um sistema distribuído é considerado *Peer-to-Peer* se os membros compartilham parte dos seus recursos. Esses recursos são necessários para prover o serviço e conteúdo oferecido pela rede e devem ser acessados pelos outros *peers* diretamente, sem passar por entidades intermediárias. Os participantes desse tipo de rede são provedores bem como requisitantes destes de serviços e conteúdos. Em P2P puro não há a existência de nenhuma entidade para prover recursos ao sistema distribuído [5].

2.1 Protocolo XMPP-Jingle

Foi concebido em 1999, o protocolo extensível de Mensagens e Presença, *Extensible Messaging and Presence Protocol* (XMPP), era anteriormente conhecido por Jabber, nome da comunidade *open-source* que iniciou o projeto como um padrão aberto para comunicação em tempo real. O desenvolvimento do XMPP foi motivado pela carência de um protocolo aberto para troca de mensagens instantâneas.

Aplicativos XMPP são capazes de oferecer os serviços de criptografia de canal, autenticação, presença, lista de contatos, mensagens um-a-um ou em grupo, notificações, descoberta de serviço, controle de fluxo e seções multimídia.

Os serviços estão definidos nas especificações publicadas pela Força Tarefa de Engenharia de Internet[2] (*Internet Engineering Task Force* - IETF), e suas extensões nos Padrões Estabelecidos XMPP[3], da *XMPP Standards Foundation*.

A extensão Jingle é usada para inicializar e administrar sessões multimídia entre duas entidades, de maneira interoperável com os padrões da Internet [3]. Devido a utilização do procotolo XMPP juntamente com a extensão Jingle, este trabalho refere-se a XMPP-Jingle sempre mencionando o uso de ambas implementações.

Stanzas são para XMMP o que o pacote é para a camada de rede do modelo OSI/ISO, ou seja, uma estrutura enviada entre os dois pontos (entidades) que estão se comunicando.

2.1.1 Como XMPP funciona

A arquitetura da rede XMMP é descentralizada e baseada em cliente/servidor, ou seja, a comunicação não é feita diretamente entre os clientes, semelhante ao serviço de e-mail. Quando um cliente de e-mail envia uma mensagem, a mesma é enviada ao servidor do destinatário, como um pacote de rede, a mensagem trafega entre os servidores de e-mail intermediários. O serviço de XMPP funciona de maneira diferente, os clientes se conectam aos servidores, que por sua vez se conectam entre si. Aqui os servidores onde os clientes estão conectados trocam dados diretamente, sem passar por nenhum outro servidor XMPP (*hop*) intermediário .

As contas dos usuários da rede XMPP são identificadas por um *JabberID* (JID) que possui a forma de um endereço de email, ou seja, *nomedousuario@servidor.tdl*, inclusive podendo ser o próprio endereço de e-mail, caso o servidor forneça ambos serviços. Um exemplo disto é o GMail da Google Inc. [4]

Um usuário pode se conectar à rede através de múltiplos dispositivos concorrentemente, a cada conexão um *resource* é designado pelo servidor, ou definido pelo usuário, para servir de identificação e roteamento das mensagens. Dessa forma, quando, por exemplo, um usuário inicia através do celular uma conversa com um amigo, o mesmo responde e esta mensagem é encaminhada para o telefone, e não qualquer outro elemento conectado na rede, com *resource* diferente.

2.2 Biblioteca libJingle

Implementação do protocolo Jingle feita pela Google Inc., distribuída em código aberto. A especificação da extensão Jingle ainda está classificada como *draft* (não concluída) e portanto a *libJingle* não provê compatibilidade com outros sistemas que se utilizam da versão atual da especificação, uma vez que o desenvolvimento da biblioteca ocorreu paralelamente com as primeiras revisões da especificação.

2.2.1 Estrutura de aplicações com libJingle

A estrutura de um programa que se utiliza da biblioteca *libJingle* possui, além da interface com o usuário, três grandes componentes[2], que são:

***XMPP Messaging Component*:** Componente de mensagens XMPP, serve como um *gateway* entre a rede (representada no diagrama pela nuvem) e as *stanzas* que chegam do Componente de Lógica e Gerenciamento da Sessão;

***Session Logic and Management Component*:** Componente de Lógica e Gerenciamento da Sessão, cuida da lógica de cada tipo de sessão, é o responsável por passar *stanzas* para o componente Ponto-a-Ponto, que cuidará dos detalhes da conexão, e depois irá ler/escrever os dados do componente para arquivos, sistema de audio ou vídeo;

***Peer-to-Peer Component*:** Componente Ponto-a-Ponto, responsável por cuidar dos detalhes da conexão entre os pontos, alocando *sockets* e monitorando a qualidade da conexão.

O tráfego de dados é realizado Ponto-a-Ponto, diretamente entre os clientes que estão trocando arquivos, por exemplo. Assim o servidor XMPP é responsável pelo envio e recebimento das *stanzas* que anunciarão a troca P2P, mas os dados não passam pela rota do servidor.

3. IMPLEMENTAÇÃO NO MIDDLEWARE GINGA

A partir do código-fonte da aplicação `pcp` que acompanha o SDK da biblioteca *libJingle* foi construído o componente e a aplicação de exemplo. A Seção 3.1 apresenta a arquitetura do *middleware* Ginga. A Seção 3.2 mostra de que forma está implementada no *middleware* o componente e a aplicação teste.

3.1 Arquitetura do Ginga

A arquitetura do *middleware* Ginga é formada por três grandes módulos, o ambiente declarativo Ginga-NCL, o ambiente imperativo Ginga-Imp e o núcleo comum Ginga-CC.

[2]http://ietf.org/ — acesso em 25/05/2010
[3]http://xmpp.org/ — acesso em 25/05/2010

[4]Os serviços de DNS são usados pelo XMPP para os fins de resolução de domínios

O Núcleo Comum do *middleware* Ginga provê as funcionalidades comuns aos outros dois ambientes e é composto pelos decodificadores de conteúdo comuns e por procedimentos de obtenção de conteúdo dos fluxos de transporte, *Transport Streams* (TS), MPEG-2 e outras redes suportadas pelo sistema operacional do receptor onde o *middleware* Ginga está instalado. Além disso Ginga-CC deve suportar o modelo de exibição, controle de acesso condicional, gerenciamento de contexto, persistência de objetos de multimídia e gerenciamento de atualizações.

Aplicações de Televisão Digital Interativa (TVDi), podem ser puramente declarativas (escritas em NCL), puramente imperativas ou híbridas. O documento inicial aceito pelo *middleware* pode ser escrito para qualquer um dos ambientes citados. Aplicações podem ser recebidas das várias redes disponíveis, entre elas o carrossel de dados da emissora, fluxo contínuo de dados pelo protocolo do padrão MPEG-2-TS, ou aplicações residentes. Os componentes do Núcleo Comum, além de serem utilizados pelos ambientes declarativo e imperativo, podem ser utilizados por aplicações vindas de qualquer fonte.

A Figura 1 mostra a Arquitetura do *middleware* Ginga com seus módulos e ambientes declarativo e imperativo.
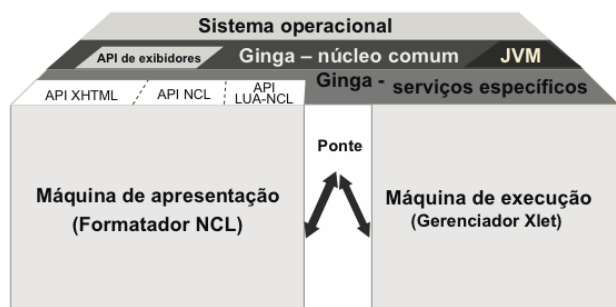
Figure 1: Arquitetura do Ginga [1]

3.2 Codificação do Componente e aplicação teste

A Figura 2 mostra o diagrama de classes do componete P2P e da aplicação de teste, um detalhe de implementação é a necessidade da aplicação herdar a classe `::br::pucrio::telemidia::ginga::core::system::thread::Thread` para que o *middleware* Ginga não trave esperando a execução da mesma.
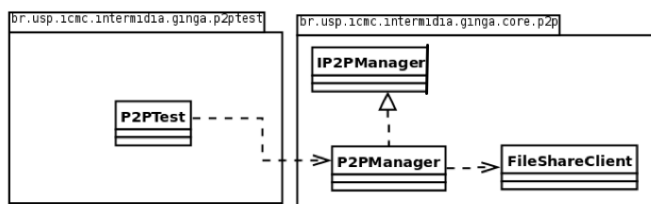
Figure 2: Diagrama de classes do componente e aplicação teste

A classe *P2PTest* possui o método `run()`, sua implementação instancia um objeto da classe *P2PManager* do componente. Essa classe tem o método *connect()*, responsável pela conexão com servidor XMPP e chamada dos métodos implementados na classe *FileShareClient*, responsável por realizar a troca de arquivos.

A classe *P2PTest* é uma estrutura *Singleton*, um padrão de projeto que restringe a criação de apenas uma instância da classe. O código-fonte de *P2PTest* mostra isso no método `getInstance()`, o cronstrutor da classe possuia rotina de *log* usada pelo *middleware* Ginga. O método `run()` configura alguns detalhes para a conexão e em seguida chama o método `connect()` da classe *P2PManager*.

P2PManager também é uma classe *Singleton*, seu método `connect()` é responsável por criar a conexão XMPP e preparar parâmetros para a instância de *FileShareClient*.

A classe *FileShareClient* implementá a extensão Jingle, a restruturação para a ampliação da API do componente se dará a partir desta classe, que concentra todo o serviço de troca de arquivos.

4. CONCLUSÕES

O aplicativo `pcp`, parte da SDK da *libJingle*, foi configurado com uma conta XMMP diferente da aplicação de teste do componente, dentro do *middleware* para a demostração de envio de um arquivo deste cliente até o *middleware* Ginga.

A execução do *middleware* Ginga imprime no terminal da máquina virtual *logs*, que foram utilizados pra confirmar a execução da aplicação, e a partir disso foi executada em outro computador o aplicativo pcp, após o estabelecimento da conexão entre os dois pontos o arquivo foi enviado com sucesso.

Todo esse desenvolvimento mostra a utilidade e aplicação do protocolo XMPP sendo usado para troca de arquivos em redes P2P, e seu uso em aplicações de TV Digital interativa.

O trabalho contribuiu para a inserção de novos recursos no âmbito da TV Digital no Brasil, o que ajudará no amadurecimento do *middleware* Ginga. O trabalho serviu também como mais um experimento da evolução que a televisão deve sofrer ao decorrer dos próximos anos, ao ser tratada como uma nova plataforma de interatividade – o que é relevante por ser a TV, atualmente, a forma dominante para acesso a informação no Brasil.

5. REFERENCES

[1] ABNT NBR 15606-2. *Televisão digital terrestre - Codificação de dados e especificações de transmissão para radiodifusão digital - Parte 2: Ginga-NCL para receptores fixos e móveis - Linguagem de aplicação XML para codificação de aplicações.*

[2] Google, Inc. *libjingle Developer Guide*, 2009.

[3] S. Ludwig, J. Beda, P. Saint-Andre, R. McQueen, S. Egan, and J. Hildebrand. Xep-0166: Jingle. *On-line*, 2009.

[4] P. Saint-AndrÈ, K. Smith, and R. TronÁon. *XMPP: The Definitive Guide: Building Real-Time Applications with Jabber Technologies.* O'Reilly Media, Inc., 2009.

[5] R. Schollmeier. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. pages 101 –102, aug 2001.

[6] L. F. G. Soares. *Ginga FrEvo & GingaRAP - Evolução do Middleware Ginga para Múltiplas Plataformas (Componentização) e Ferramentas para Desenvolvimento e Distribuição de Aplicações Declarativas. Edital CTIC*, 2009.